CN 53-1189/P ISSN 1672-7673

基于 Web.py 的 MUSER 软件系统界面实现*

蔡能健[1]，刘东浩[2]，邓 辉[1]，卫守林[1,3]，王 锋[1,3]，
梅 盈[1,3]，戴 伟[1,3]，刘应波[4]，吴静平[5]
(1. 昆明理工大学云南省计算机技术应用重点实验室，云南 昆明 650500；2. 中国科学院国家天文台，北京 100012；
3. 中国科学院云南天文台，云南 昆明 650011；4. 云南省科学技术情报研究院，云南 昆明 650051；
5. 云南省信息技术发展中心，云南 昆明 650034)

摘要：明安图射电频谱日像仪(Mingantu Ultrawide SpEctral Radioheliograph，MUSER)已经完成硬件建设，与之配套的数据处理系统也初步开发完成。为提高用户操作的可用性，简化操作难度，数据处理系统在提供传统的命令行用户界面的同时，也提供了网络交互方式。简单介绍了 Web.py 的基本概念和内部组件交互流程，讨论了基于 Web.py 进行网络开发的方法和流程，并利用 Web.py 框架实现了数据处理系统的网络界面交互。成果已经应用，提高了日像仪数据处理系统的可用性，也为其它天文软件系统的开发提供了一定的借鉴与参考价值。

关键词：天文软件系统；Web.py；用户界面
中图分类号：TP311.1 **文献标识码**：A **文章编号**：1672-7673(2017)02-0229-07

为推动国内太阳射电天文观测技术的发展，我国天文学家提出建设的明安图射电频谱日像仪已经全面建成[1]。该设备可以在厘米-分米波段对太阳同时进行高空间、高时间和高频率观测，观测所得数据可以更好地研究太阳活动。项目分两期建设，第 1 期是 MUSER-I 低频阵(0.4~2 GHz)，由 40 个 4.5 m 的天线组成，在 64 个频率上成像；第 2 期 MUSER-II 高频阵(2~15 GHz)，由 60 面 2 m 的天线组成，在 528 个频率上成像。

明安图射电频谱日像仪的数据处理系统是整个项目的关键组成部分，设计并实现快速、可靠、方便的数据处理软件已成为发挥望远镜作用的重要支撑。当前基于浏览器/服务器(B/S)模式开发网络应用软件已经成为软件开发的主流，一方面，这一类软件具有高移植性的特点；另一方面，网络应用软件无需部署，对后期维护非常有利。从长远建设来看，实现一个具有友好人机交互的网络界面，集成相应的数据处理程序，满足现场观测以及远程数据处理的要求是非常必要的。

开发网络应用可用的技术很多，最为常见的是基于 J2EE 或者.NET 技术，分别采用 Java 与 C#语言开发。一般来说，这一类软件存在部署运行环境复杂、开发周期较长等不足。近年来，天文软件开发大多采用 Python 语言，明安图射电频谱日像仪的数据处理系统也是基于 Python 开发完成。为了更好地和数据处理系统集成和交互，明安图射电频谱日像仪的网络用户界面同样采用 Python 语言开发。本文首先介绍了数据处理系统的架构及操作界面的功能，接着讨论了轻量级网络框架 Web.py 的内部组件交互流程，重点讨论了使用 Python 语言和 Web.py 进行快速开发的方法，以及如何利用 Web.py 框架实现网络界面与功能封装。

1 明安图射电频谱日像仪数据处理系统

随着望远镜性能的显著提升，产生的数据成倍增长，处理过程日趋复杂，计算机软件系统已成为

* 基金项目：国家自然科学基金（U1231205，11403009）资助.
收稿日期：2016-06-22；修订日期：2016-07-09
作者简介：蔡能健，男，硕士. 研究方向：计算机应用技术. Email：cainengjian@cnlab.net
通讯作者：邓 辉，女，教授. 研究方向：科学数据处理. Email：denghui@cnlab.net

天文数据处理中不可或缺的工具。天文软件系统可以分为数据处理和数据发布两大类。数据处理软件一般部署于本地，采用单机或计算机集群方式完成观测数据的处理，包括最为常见的 IDL、CASA、AIPS 等。数据发布软件一般基于网络应用和表述性状态传递(Representational State Transfer，REST)风格的网络服务发布技术，如虚拟天文台中的锥形检索等。

在明安图射电频谱日像仪的日常数据处理中，包含了两方面的需求，一是对历史数据的处理，另外一个是每 5 秒钟的采样观测数据的实时处理。为了满足高速数据处理的需求，开发和实现了一套部署于明安图观测站的高性能计算系统[2]。同时为了天文学家离线处理的需要，开发了一套可运行在单机上的客户端软件，实现与 CASA 类似的功能。日像仪软件系统整体功能结构如图 1，主要包括 3 部分，分别用于核心数据处理、用于交互的用户界面以及数据库。核心数据处理模块采用 Opencluster[3] 框架，支持分布式任务调度。核心的底层数据获取、数据处理(包括 CLEAN、FFT、IFFT 等)、数据存储(UVFITS 等)等功能已经基于 Python 开发完成[4-5]，并通过 Opencluster 实现整个计算资源的分配和调度。软件系统采用 MySQL 数据库存储数据处理中的若干关键参数，如光纤传输时延、气象数据等[6]，同时为了提高查询速度，也支持将数据导出到 Redis 内存数据库。用户界面主要实现数据处理任务提交、天线状态维护、自动气象站和卫星数据修正配置等功能。

目前，明安图射电频谱日像仪已经实现了与 CASA 操作类似的命令行(CLI)接口[7]，但命令行方式存在命令多、表现方式不直观等不足。为了进一步提高数据处理系统的使用效率，使用 Web.py 开发了更为友好的数据处理系统交互界面。

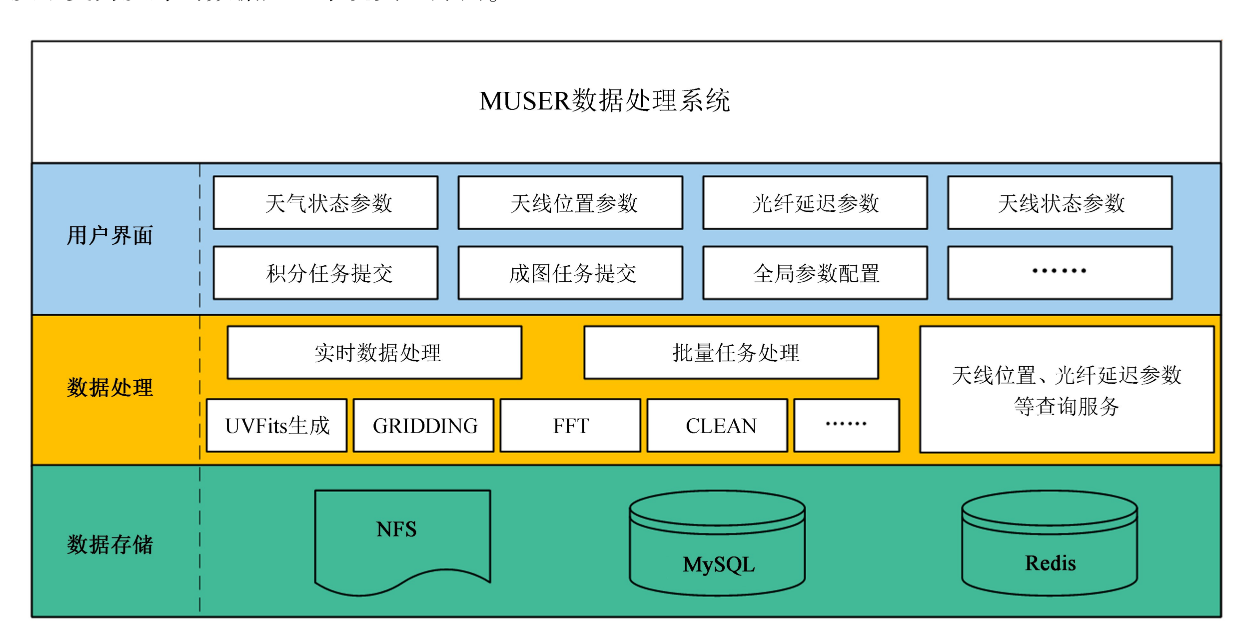


图 1 明安图射电频谱日像仪数据处理系统结构图

Fig. 1 The structure chart of MUSER data processing system

2 Web.py 框架分析

Web.py[8]是一个纯 Python 实现的轻量级网络应用开发框架，支持模型-视图-控制器(Model View Controller，MVC)模式开发，可通过简单的编程将普通的 Python 类映射为请求处理对象，原生支持编写 REST 架构风格的网络服务，当前的版本是 0.38。Web.py 提供了最小化的网络服务器，可以作为一个独立的守护进程运行，或者借助于网络服务器网关接口(Web Server Gateway Interface，WSGI)在任何兼容它的服务器上运行。

Web.py 不依赖于其他第三方 Python 库，可以下载官方的二进制包(http://webpy.org/static/web.py-0.38.tar.gz)，解压缩后执行 python setup.py install 命令进行安装，也可以通过 easy_install 或 pip 方式安装。

Web.py 框架执行过程主要在 application.py、wsgi.py、httpserver.py 和 wsgiserver 目录的模块初始化文件_init_.py 这 4 个文件中定义。application.py 是一个网络应用运行的入口；wsgi.py 是将网络应用运行在内置的简单的网络服务器，或者兼容 WSGI 标准的服务器(如 FastCGI)；httpserver.py 是内置的网络服务器，符合 WSGI 规范；wsgiserver 目录中的_init_.py 文件负责创建服务器，接收客户端请求，并响应请求。从浏览器访问开始，Web.py 的内部执行流程如图 2。

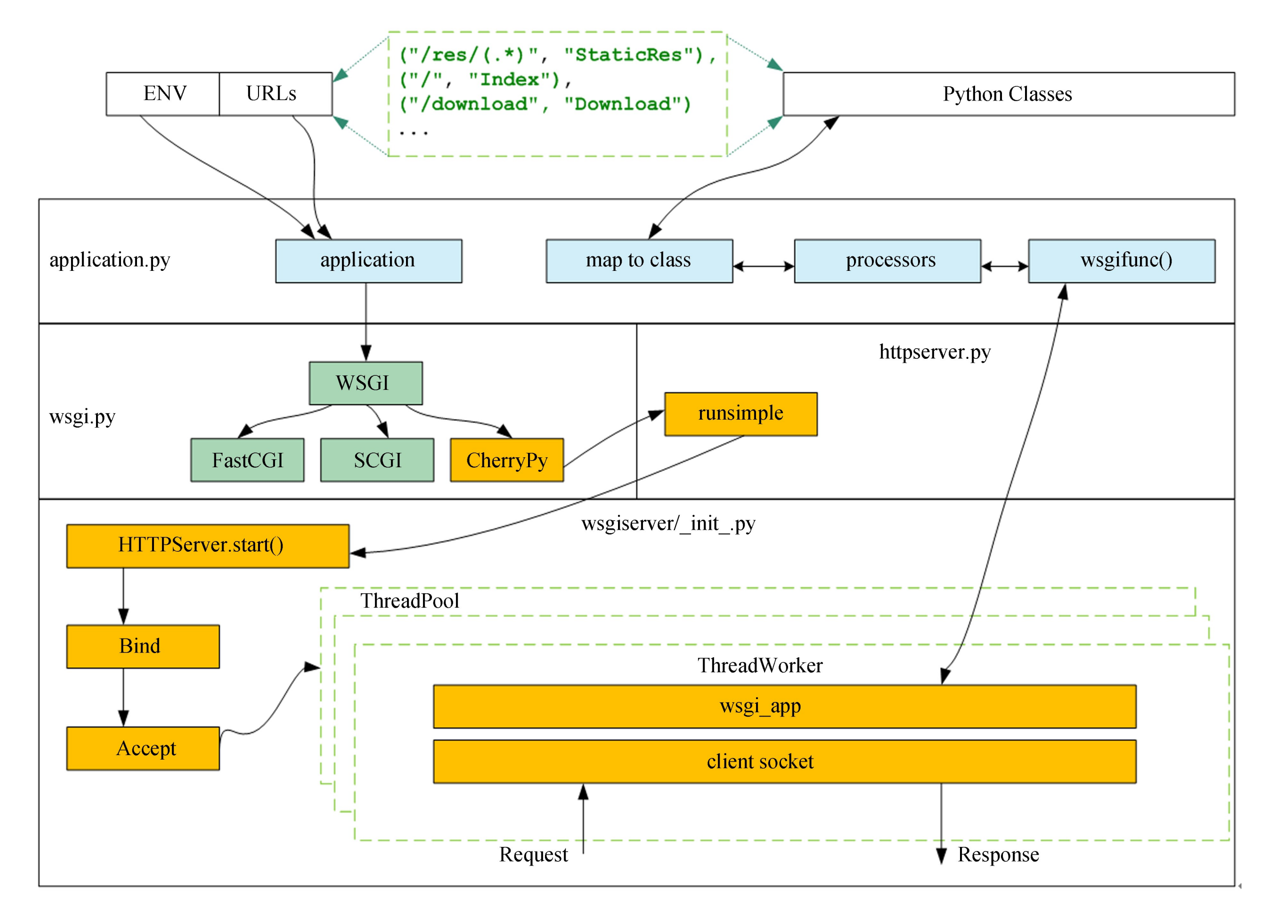


图 2　Web.py 组件图

Fig. 2　The component diagram of Web.py

Web.py 中网络应用在构建时，需传入统一资源定位符(Uniform Resource Locator，URL)，它是访问当前网络应用的相对路径和负责处理该路径请求的 Python 类的一个二元组的集合。每当浏览器发出新的请求，Web.py 将请求的路径和 URLs 中路径做匹配，最终将请求代理到定义的 Python 类中相应的方法执行。应用启动时，在 wsgi.py 文件中根据环境变量配置不同，分别启动不同的 WSGI 处理器(FastCGI、flup、内置简单网络服务器)。启动内置网络服务器时，首先调用 httpserver. py 中的 runsimple 方法，runsimple 方法再调用 WSGIServer 方法，WSGIServer 中创建使用了 CherryPy 的基于线程池的 WSGI 服务器，其类的定义为 CherryPyWSGIServer。

CherryPyWSGIServer 的构造函数中，执行初始化线程池、WSGI 网关等操作。启动服务器时执行 CherryPyWSGIServer 的基类 HTTPServer 中的 start 方法，在指定地址和端口监听客户端请求，具体的请求处理，由执行请求任务的 WorkerThread 线程处理。

3　交互界面设计

数据处理系统用户界面需要实现天线状态、天线位置、气象条件、光纤延迟等数据维护及数据处理任务提交等功能。本节重点介绍使用 Web.py 开发过程中几个关键技术。

3.1 URL 设计与处理

就一个网络应用而言，URL 的组织是最重要的部分，良好的 URL 设计能清晰地表达整个网络应用软件的控制思路。Web.py 中 URL 的定义支持 RESTful 的格式，RESTful 的 URL 具有结构清晰、符合标准、易于理解、扩展方便等优点，因此网络交互界面的 URL 也使用 RESTful 设计。

REST 中包含了两个重要概念，一是资源，二是状态转移。资源代表了操作对象，通常是名词，命名方式一般和数据库的表名对应。表 1 列出了网络交互界面的部分类和资源的对应关系。状态转移代表了对资源的操作导致资源的状态发生变化，具体到 HTTP 协议中就是 4 个表示操作方式的动词：GET，POST，PUT 和 DELETE，正好对应到增、删、查、找(Create，Read，Update，Delete，CURD) 4 种数据操作。在 Web.py 框架实现 GET，POST，PUT，DELETE 方法，只需要在相应的处理类中定义同名的函数即可。系统主要用到 GET 和 POST 两种请求，GET 请求用于获取数据，POST 请求主要用于插入或者修改数据时提交新的表单请求。

表 1 网络交互界面的类和资源的对应关系

Table 1 Mappings of classes to resourcesin MUSER web design

URL	Python 类	说明
/	Index	首页
/antenna	Antenna	天线
/antenna/position	AntennaPosition	天线位置
/antenna/flag	AntennaFlag	天线状态
/antenna/delay	AntennaDelay	光纤延迟
/parameter	Parameter	系统参数
/job	Job	计算任务

Web.py 的 URL 控制模式简单、灵活。网络应用启动时，需要将所有 URL 和 Python 类的映射关系统一定义在一个元组列表中，元组的格式为：(“URL 相对路径”，“处理类”)，例如表 1 中对天线的操作，使用的路径是“/antenna”，处理类是“Antenna”，则元组的定义为(“/antenna”，“Antenna”)。

此外，Web.py 提供了强大的正则表达式设计更灵活的 URL 路径，例如查询一个计算任务状态的 URL 可以定义为“/job/(.+)”，在/job/后面的内容会被捕捉，然后作为参数被用在 GET 或 POST 方法中。假设一个计算任务 ID 为 10，可以使用“/job/10”查询其执行状态。“/job/10”能成功匹配“/job/(.+)”，在具体查询的处理类中可以获得这个计算任务 ID，再执行具体的查询，并返回结果。

3.2 数据库操作

Web.py 支持多种数据库操作。Web.py 封装了各种数据库的操作，支持事务，提供了简单易用的应用程序编程接口，如表 2。具体和数据库的底层操作需要借助于相应数据库操作模块，如 MySQL 的 Python 库 MySQLdb，SQLServer 的 Python 库 pymssql。Web.py 的数据库部分的相关代码在 db.py 文件中，入口函数是 database( )。根据 dbn(数据库名)从 dict 里查找对应类型数据库的实现类，再用其余参数初始化这个数据库实现类，返回具体的数据库操作对象，使用这个对象实现对数据库表执行增、删、改、查的操作。

表 2 Web.py 数据库操作方法

Table 2 Methods of database operations in Web.py

方法	功能	实例
insert	新增	db.insert (self，表名，变量 1，变量 2，…)
query	查询	db.query (self，表名，变量 1，变量 2，…)
update	更新	db.update (self，表名，变量 1，变量 2，…)
delete	删除	db.delete (self，表名，变量 1，变量 2，…)

数据处理系统用到了 MySQL 和 Sqlite 数据库，以及 Redis 内存存储。其中 MySQL 作为主数据库支持读写操作，Sqlite 在单机版处理系统使用，Redis 主要在分布式环境中使用。系统定时将 MySQL 的数据导出到 Sqlite 和 Redis，Sqlite 和 Redis 在数据处理系统中提供只读操作，以保障数据的一致性。Redis 是一个基于内存键值存储系统，支持高速高并发的读写操作，为明安图射电频谱日像仪的分布式处理环境气象、天线相关参数的高速访问提供了保障。Redis 不是关系型数据库，它的访问和操作需要使用 Python 的 redis 库。

3.3　模板渲染

网络框架都提供模板技术实现复杂的网络页面，Web.py 也不例外。Web.py 的模板功能实现的源码在 template.py 文件中。Web.py 的模板语言使得可以在模板文件 HTML 中使用 Python 语言的强大功能，每一个 HTML 文件必须在第 1 行通过 $ def with( ) 定义在该文件中用到的所有变量，这些变量名需要和 Python 中返回模板文件的变量命名一致，在 HTML 中就可以直接使用这些变量。

Web. py 模板的强大之处除了引用 Python 中变量，还支持流程控制，语法与 Python 类似。控制结构语句前面需要加上“ $ ”，可以实现 for、while、if、elif 和 else 等语句。如明安图射电频谱日像仪中的任务处理列表的部分模板如下：

```
$ for job in jobs://jobs 为任务列表变量
$ job.status//输出单个任务的状态
```

Web.py 的模板支持在 Python 中 Render 渲染模板文件，也可以直接渲染 HTML 的字符串。如显示气象数据列表，需要匹配 weather.html 模板，使用如图 3 的代码。

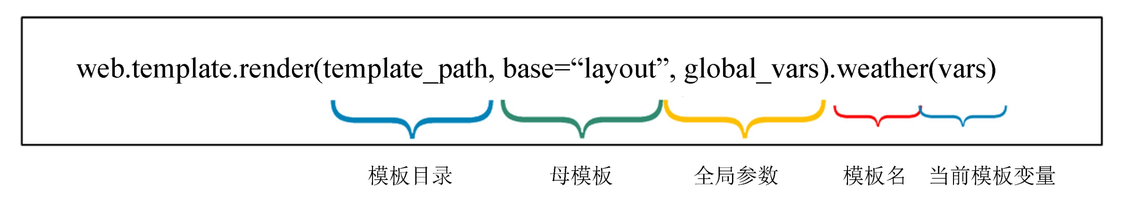


图 3　Web.py 模板渲染说明

Fig. 3　Specification of template-rendering in Web.py

明安图射电频谱日像仪中使用一个专门的模板文件夹 templates，存放所有的模板文件。响应客户端请求时，Web.py 从模板目录寻找模板文件，匹配相应的模板文件后，渲染这个模板文件并返回浏览器。Web.py 支持模板嵌套，可以使用 base 指定母模板。

4　系统界面

基于 Web.py 开发的数据处理系统用户界面已基本开发完成并应用到实际数据处理中，前端使用 Bootstrap 构建了响应式页面，系统界面如图 4。系统实现了在命令行界面配置系统参数，为多方面数据(原始数据、异常天线信息、仪器状态、天气)的录入、维护、查询和使用提供了方便，确保了后续数据处理的有效性和可靠性。同时，网络界面提供的数据处理参数的配置，方便了复杂任务的提交，与命令行界面相比，具有更好的交互性和可用性。

5　结束语

本文介绍了基于 Web.py 框架进行明安图射电频谱日像仪数据处理系统用户界面开发的过程。通过 Web.py 轻量级框架提供的核心模块可以根据需求快速开发出一个稳定有效的网络应用程序。Web.py

具有轻巧简单、代码设计风格简洁的特点。实践证明，采用 Web.py 开发的程序稳定可靠，能够满足日像仪数据处理系统中用户界面开发的需求。目前，系统初步搭建完毕，但是仍然有不足的地方，例如更可靠地数据错误校验，数据编辑用户友好性提升，以及提供更多的实时交互功能，这些将在后续工作中继续研究和探讨。

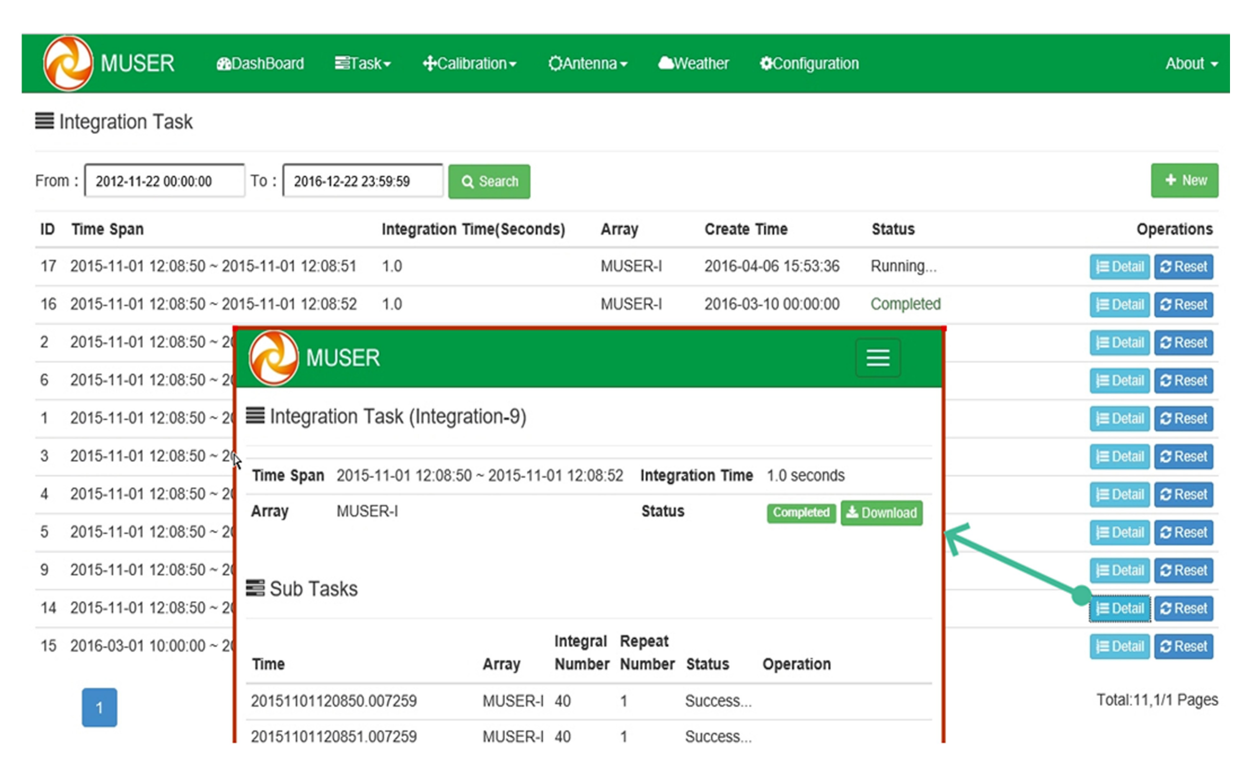


图 4 明安图射电频谱日像仪数据处理系统用户界面

Fig. 4 The user interface of MUSER data processing system

参考文献：

[1] 颜毅华，张坚，陈志军，等. 关于太阳厘米-分米波段频谱日像仪研究进展［J］. 天文研究与技术——国家天文台台刊，2006，3(2)：91-98.
Yan Yihua，Zhang Jian，Chen Zhijun，et al. Progress on Chinese radioheliograph in cm-dm wavebands［J］. Astronomical Research & Technology——Publications of National Astronomical Observatories of China，2006，3(2)：91-98.

[2] Wang Feng，Mei Ying，Deng Hui，et al. Distributed data-processing pipeline for Mingantu Ultrawide Spectral Radioheliograph［J］. Publications of the Astronomical Society of the Pacific，2015，127(950)：383-396.

[3] Wei Shoulin，Wang Feng，Deng Hui，et al. OpenCluster：a flexible distributed computing framework for astronomical data processing［J］. Publications of the Astronomical Society of the Pacific，2017，129(972)：1-14.

[4] 马荣庭，刘飞，邓辉. CSRH-I 观测数据接收子系统的设计与实现［J］. 天文研究与技术，2015，12(1)：63-69.
Ma Rongting，Liu Fei，Deng Hui. The design and implementation of a data-receiving subsystem of the CSRH-I［J］. Astronomical Research & Technology，2015，12(1)：63-69.

[5] 梅盈，刘东浩，王锋，等. 中国频谱射电日像仪 FITS-IDI 文件格式研究［J］. 天文研究与技术——国家天文台台刊，2014，11(4)：388-395.

Mei Ying, Liu Donghao, Wang Feng, et al. A study of the FITS-IDI format for the Chinese Spectral Radio Heliograph [J]. Astronomical Research & Technology——Publications of National Astronomical Observatories of China, 2014, 11(4): 388-395.

[6] 卫守林, 石聪明, 高姣姣, 等. Vantage Pro 气象站实时数据采集与在 MUSER 中的应用研究 [J]. 天文研究与技术, 2016, 13(1): 117-123.
Wei Shoulin, Shi Congming, Gao Jiaojiao, et al. A study on the real-time collection of the Vantage Pro weather station and the application in the MUSER [J]. Astronomical Research & Technology, 2016, 13(1): 117-123.

[7] 陈蒙, 王锋, 邓辉, 等. 基于 Python 的天文软件命令行界面设计与实现 [J]. 天文研究与技术, 2015, 12(2): 196-203.
Chen Meng, Wang Feng, Deng Hui, et al. Design and implementation of an astronomical command line interface system based on the Python [J]. Astronomical Research & Technology, 2015, 12(2): 196-203.

[8] Swartz A. Web.py Install Guide [EB/OL]. http://webpy.org/install.

The Application Method of Web.py in Developing MUSER Software System

Cai Nengjian[1], Liu Donghao[2], Deng Hui[1], Wei Shoulin[1,3], Wang Feng[1,3],
Mei Ying[1,3], Dai Wei[1,3], Liu Yingbo[4], Wu Jingping[5]

(1. Key Laboratory of Applications of Computer Technology of the Yunnan Province, University of Science and Technology of Kunming, Kunming 650500, China, Email: denghui@cnlab.net; 2. National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China; 3. Yunnan Observatories, Chinese Academy of Sciences, Kunming 650011, China;
4. Yunnan Academy of Scientific & Technical Information, Kunming 650051, China;
5. Yunnan Information Technology Development Center, Kunming 650034, China)

Abstract: The hardware construction of Mingantu Ultrawide SpEctral Radioheliograph (MUSER) has been completed, and data processing system for the MUSER project has also been completed. In order to improve its user operation availability, simplify its operation difficulty, MUSER data processing system provides both the traditional command-line interface and the WEB interface for the users. This paper introduced a framework named web.py and discussed how to apply the framework to quickly develop a WEB interface for MUSER. Overall, for astronomy software written in Python, web. py is a proper choice in developing its WEB interface.

Key words: Astronomy software; Web.py; User interface